

AD-A086 778

NAVAL RESEARCH LAB WASHINGTON DC
SYNTHESIS OF MCJ CODES.(U)
APR 80 H M BECK

F/G 9/4

UNCLASSIFIED

NRL-8393

SBIE-AD-E000 444

NL

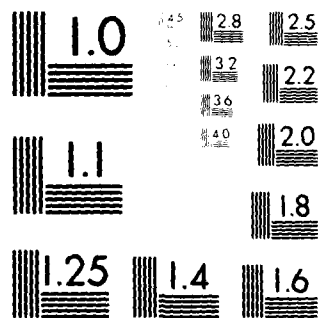
1-2 /
AD
8-18-80

END

DATE
FILMED

8-80

DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS-1963-A

ADA 086778

LEVEL III

AD 7E 500 444
NRL Report 8393

(12)

Synthesis of MCJ Codes

HUGO M. BECK

*Systems Integration and Instrumentation Branch
Communications Sciences Division*

April 4, 1980



DTIC
ELECTE
JUL 17 1980
S D D

DDC FILE COPY

✓ NAVAL RESEARCH LABORATORY
Washington, D.C.

Approved for public release; distribution unlimited.

80 5 19 208

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER NRL Report 8393	2. GOVT ACCESSION NO. ✓ AD-A086778	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) SYNTHESIS OF MCJ CODES		5. TYPE OF REPORT & PERIOD COVERED Interim report on a continuing NRL problem
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Hugo M. Beck		8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS Naval Research Laboratory Washington, DC 20375		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS 11402N; X-0793-SB; 0; 75-0115-0-0
11. CONTROLLING OFFICE NAME AND ADDRESS Naval Electronic Systems Command PME 117-201 Washington, DC 20360		12. REPORT DATE April 4, 1980
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office)		13. NUMBER OF PAGES 20
		15. SECURITY CLASS. (of this report) UNCLASSIFIED
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE
16. DISTRIBUTION STATEMENT (of this Report) Approved for public release; distribution unlimited.		
17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)		
18. SUPPLEMENTARY NOTES		
19. KEY WORDS (Continue on reverse side if necessary and identify by block number) MCJ codes Error control Maximum distance separable Optimal codes		
20. ABSTRACT (Continue on reverse side if necessary and identify by block number) Constructive techniques are given for the encoding and decoding of Massey-Costello-Justesen (MCJ) error control block codes. These codes form a family of maximum distance separable (MDS) codes distinct from the Reed-Solomon (RS) codes. The MCJ codes offer ease of encoding and decoding for either hardware or software implementations, based on the more natural Galois ground field mathematical operations involved in the associated algorithms. Because of inherent burst error control capabilities, these codes can be considered for channel error protection as well as for end-to-end message error control.		

DD FORM 1 JAN 73 1473

EDITION OF 1 NOV 65 IS OBSOLETE
S/N 0102-014-6601

UNCLASSIFIED
SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

CONTENTS

INTRODUCTION	1
SYNTHESIS	1
CORRECTION PROCEDURE	8
Specific Example	8
DECODING ALGORITHM	11
SIMPLIFIED SYNTHESIS	17
CONCLUSIONS	17
ACKNOWLEDGMENT	17
REFERENCES	18

Accession For	
NTIS GRA&I	<input checked="" type="checkbox"/>
DDC TAB	<input type="checkbox"/>
Unannounced	
Justification _____	
By _____	
Distribution/	
Availability Codes	
Dist.	Avail and/or special
A	

S DTIC D

ELECTE

JUL 17 1980

D

SYNTHESIS OF MCJ CODES

INTRODUCTION

It has become increasingly important to employ an effective mechanism for error control in the initial design or redesign of communication systems, both for channel protection and for end-to-end use. The most efficient codes for these applications are the maximum distance separable (MDS) codes, and the intent here is to present constructive techniques for the implementation of Massey-Costello-Justesen (MCJ) codes, an important class of MDS codes.

SYNTHESIS

The purpose of this study was to improve our understanding of constructive techniques for the engineering implementation of systematic block codes in the Hamming metric, in particular the MCJ codes [1]. The MCJ codes of immediate interest are of code length a prime number p . Most prime number alphabet sizes commensurate with record communication requirements and with reliable computer to computer communication requirements appear to be satisfied by primes $p < 10^4$.

The procedures discussed here are quite general, but the description is given in terms of a specific example using the prime number $p = 37$ in the finite field $GF(37)$. This is the smallest prime number that can contain a full numeralphabet including the 26 letters, 10 numerals, and a spacing symbol. These MCJ prime number length codes have superior error-correcting properties because they are MDS codes, can be decoded in a straightforward manner using well-defined algebraic techniques including the theory of algebraic and geometric invariants not in general available to Bose-Chandhuri-Hocqueuhem (BCH) code structures, and can be chosen to closely match communication circuit and user requirements over real channels.

As BCH codes are defined by generating polynomials with distinct roots, and as the most useful forms of MCJ codes are generated by a repeated root, the usual BCH synthesis techniques prove to be inadequate for these codes. However, many decoding techniques usually associated with BCH codes can be used to decode MCJ codes [2-4].

Note especially that the MCJ codes imply the removal of the usual restriction on cyclic codes that the code length n and the characteristic p be relatively prime, usually indicated by $(n, p) = 1$, where $q = p^m$. This gives the MCJ codes a slight coding advantage over the other optimal or MDS codes, the Reed-Solomon (RS) codes.

To demonstrate the general principles involved, the code to be discussed has generating polynomial $g(x) = (x - c)^n - k$ and will be described in the cyclic case where the constant $c = 1$, the code length $n = 37$, and the number of information symbols $k = 31$.

Manuscript submitted January 11, 1980.

The starting point here is with the Vandermonde matrix, which in the multiple root case is a generalization of the Pascal triangle. This matrix $V(c)$ for $c = 1$ gives the powers $(x - 1)^i$, $0 \leq i \leq 36$. Figure 1 emphasizes its relationship to the system function S_0 , as it evolves.

Figure 2 shows the location of the basis vectors for the row space that defines all the code vectors of the (37, 31) code. The code generator matrix $G_1(x)$ is developed from multiples $(x - c)^i g(x)$ instead of $x^i g(x)$ as in the usual synthesis of cyclic codes. To be noted specifically in Fig. 2 are powers $(x - 1)^i$, $31 \leq i \leq 36$, which when read as vertical vectors from left to right also are the truncated powers of $(x + 1)^j$, $0 \leq j \leq 36$, and which are in the form of a matrix $H_1(x)$ whose rows are generated by $(x - 1)^i h(x)$, and $h(x) = (x - 1)^n / g(x) = (x - 1)^{31}$.

The matrix S_1 in Fig. 3 contains the reduced echelon $[I_k P]$ as derived by elementary row operations on G_1 of Fig. 2. The transmit system matrix $S_0 = S_1^T$ is shown in Fig. 4, which defines the systematic MCJ code. The decoding receive system matrix S_0^{-1} shown in Fig. 5 satisfies $S_0 S_0^{-1} = I$, where I is the 37×37 identity matrix. The matrix generates syndromes exactly as in Fig. 2 of reference [1], but physically in an entirely different manner, even though both are implementations of $(x + 1)^j$. It is interesting to note that the physical realization of the preceding encoders and decoders, when implemented by linear (or non-linear) sequence generators, is more naturally described by the canonical form given by Bose and Chaudhuri in their original paper [3], than by Peterson and Weldon [4]. However, the Massey virtual encoder [1] is best described by Figs. 4 and 5 of this report.

Several different equivalent definitions for MCJ codes are possible as approached by different points of view, but no attempt is made here to exhaust these possibilities. Finally, although the MCJ codes will effectively correct random errors, they also have inherent burst error correction capabilities.

An important consideration in the synthesis of block codes is the inclusion of fail-safe error correction procedures in the decoding algorithm so that catastrophic error correction or error propagation is minimized. The following theorem is included as one way of accomplishing this; however, the most productive test is outlined in paragraph 4.0 of the decoding algorithm.

Theorem: If the homogeneous matrix of coefficients is of rank r , the nonhomogeneous matrix or system of coefficients is consistent provided that the rank of this augmented matrix is also r .

Consistency tests in the decoding procedure are pursued under five hypotheses as follows:

- H_4 : Four or more errors
- H_3 : Three errors
- H_2 : Two errors
- H_1 : One error
- H_0 : Zero errors.

V(1)=

Fig. 1 -- Generalized Vandermonde matrix

52

[illegible]

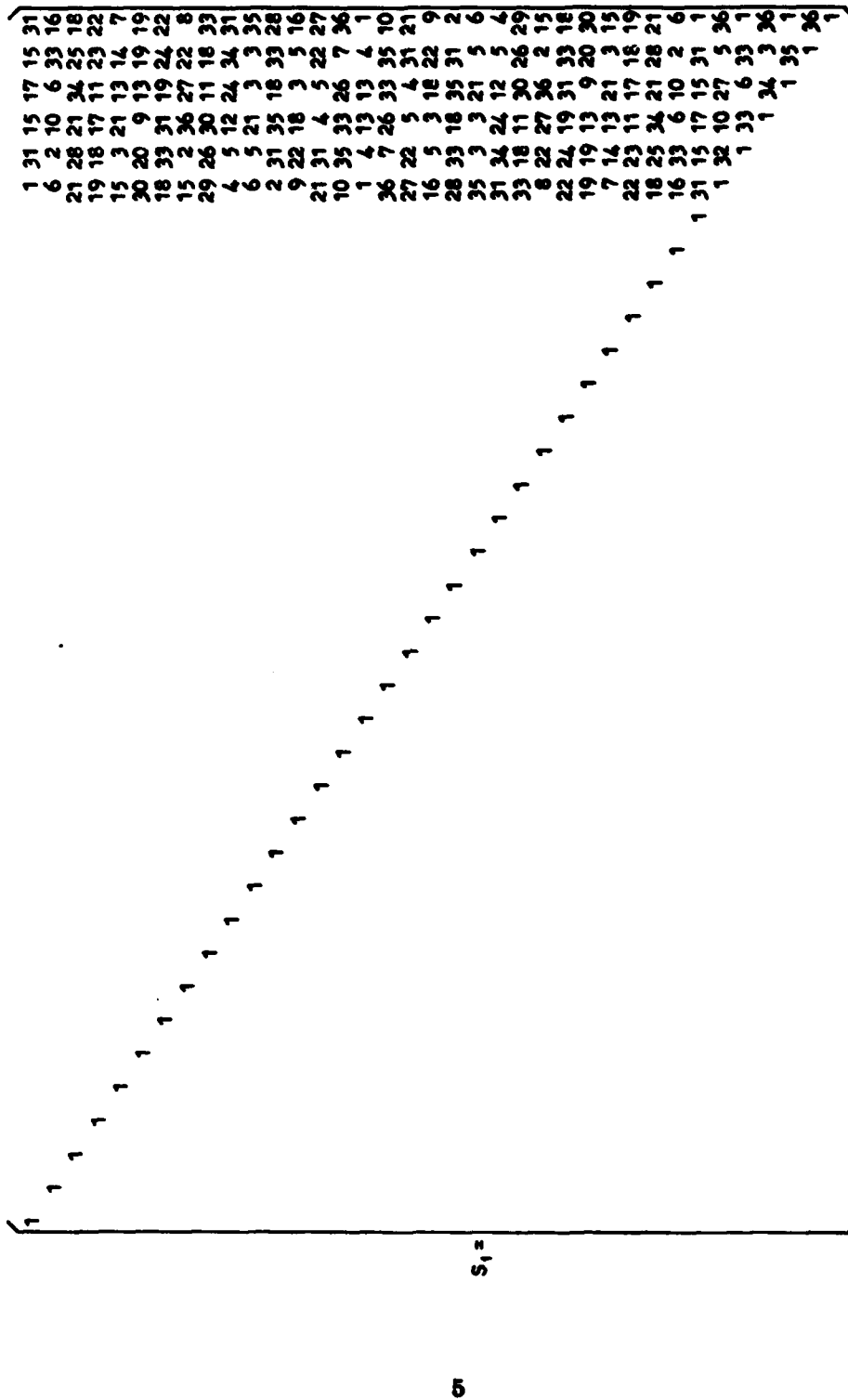


Fig. 3 — Reduced echelon matrix

BECK

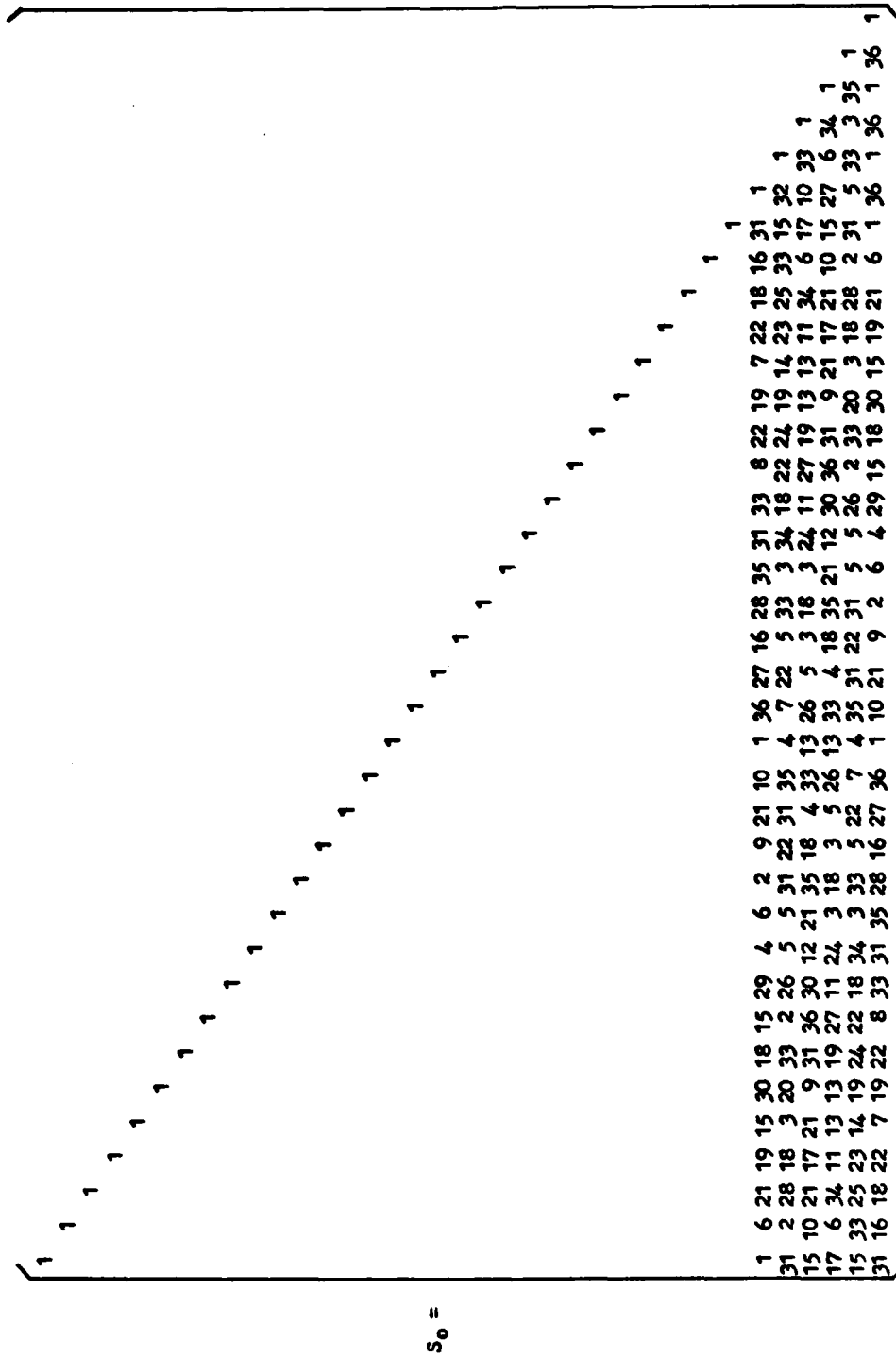


Fig. 4 — Transmit system matrix

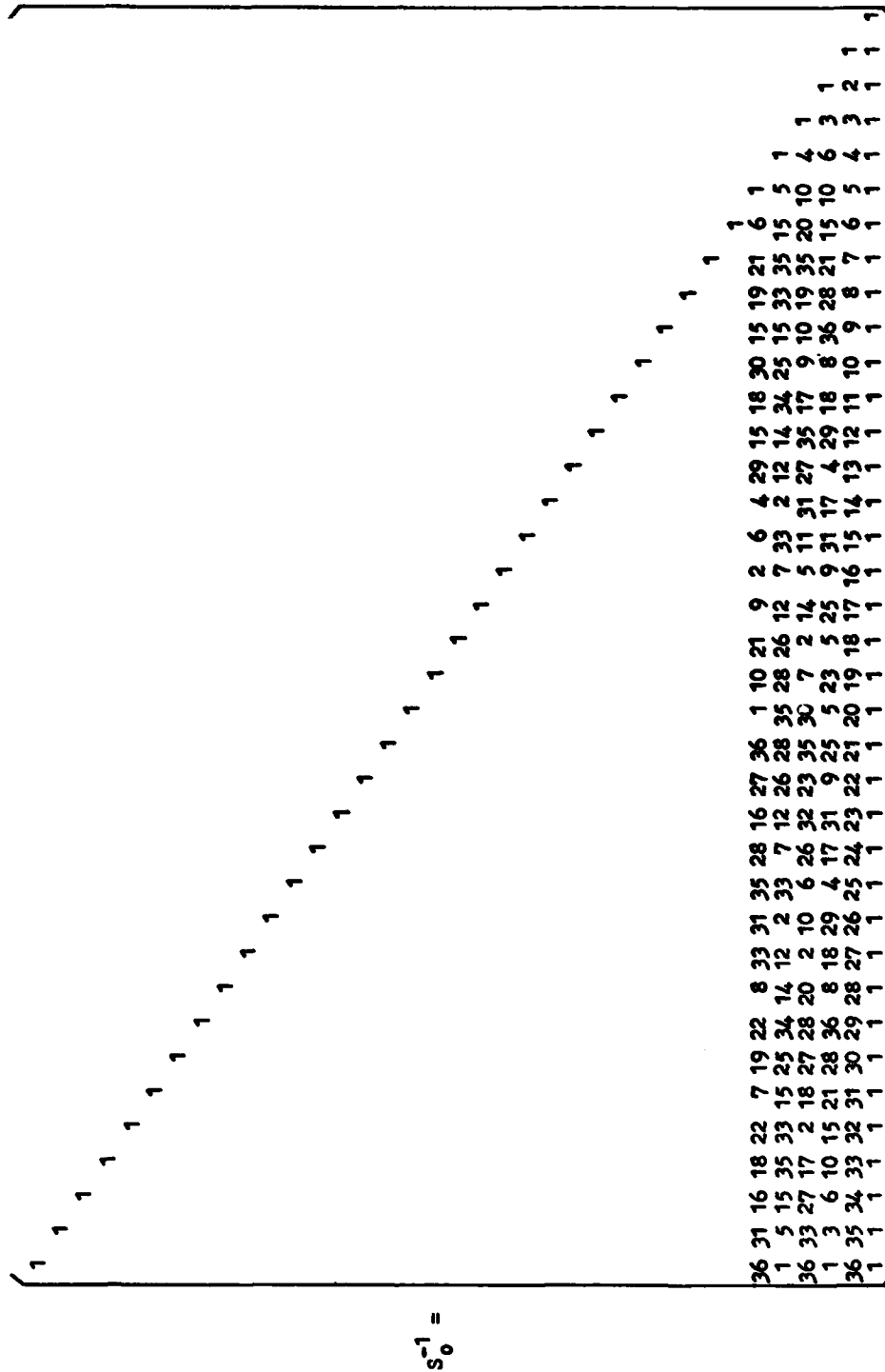


Fig. 5 - Receive system matrix

BECK

The specific purpose here [2] is to distinguish between cases where $\leq t$ errors occur, and cases where $> t$ errors occur and correction should not be attempted.

CORRECTION PROCEDURE

Preliminary definitions:

Let

$$d = n - k + 1 = 2t + 1$$

$$m_0 = 0$$

$$m = 1$$

$$n = p^m = p = \text{code length}$$

$$k = \text{number of information digits}$$

$$n - k = \text{number of parity digits}$$

$$t = \text{maximum number of correctable errors}$$

Transmitted signal

$$f(x) = f_0 + f_1x + \dots + f_{n-k-1}x^{n-k-1} + \dots + f_{n-1}x^{n-1}.$$

Noise signal

$$e(x) = e_0 + e_1x + \dots + e_{n-k-1}x^{n-k-1} + \dots + e_{n-1}x^{n-1}.$$

Received signal

$$r(x) = f(x) + e(x);$$

$$r(x) = r_0 + r_1x + \dots + r_{n-k-1}x^{n-k-1} + \dots + r_{n-1}x^{n-1}.$$

The syndrome

$$s(x) = s_0 + s_1x + \dots + s_{n-k-1}x^{n-k-1};$$

$$g(x) = (x - c)^{p-k}.$$

Specific Example

The following procedure is adequate for the decoding of any MCJ code described here and uses a specific example in the Galois field $GF(p)$ with $p = 37$, and the following param-

eters: $c = 1$, $n = 37$, $k = 31$, $n - k = 7$, and $t = 3$, which will correct up to three character errors. The block diagram to be followed is given in Fig. 6.

The received signal sequence is entered into the box $r(x)$, following which the syndromes s_i are calculated in the syndrome former, more explicitly detailed in Fig. 7. The modified syndromes S_i in Fig. 6 are calculated [5] as shown in Fig. 8 as a matrix multiplication derived as follows:*

$$S_i = \sum_{j=1}^i \left\{ \begin{matrix} i \\ j \end{matrix} \right\} j! c^j s_j, \quad 1 \leq i < p - k$$

$$S_0 = s_0$$

Where also

$$S_i = \sum_{r=1}^{t-j} y_r x_r^i \quad 0 \leq i < p - k$$

Then define, for $j = 0, 1, 2, \dots, t - 1$, the Newton matrix N_{t-j} , and the Prony matrix P_{t-j} :

$$N_{t-j} = \begin{bmatrix} S_0 & S_1 & S_2 & \cdots & S_{t-j-1} \\ S_1 & S_2 & S_3 & \cdots & S_{t-j} \\ S_2 & S_3 & S_4 & \cdots & S_{t-j+1} \\ & & & \ddots & \\ & & & & S_{2(t-j-1)} \end{bmatrix}$$

$$P_{t-j} = \begin{bmatrix} x_0^{m_0} y_0 & 0 & 0 & \cdots & 0 \\ 0 & x_1^{m_0} y_1 & 0 & \cdots & 0 \\ 0 & 0 & x_2^{m_0} y_2 & \cdots & 0 \\ & & & \ddots & \\ & & & & x_{t-j-1}^{m_0} y_{t-j-1} \end{bmatrix}$$

*An error in Eq. (32) in Ref. 1 is corrected here.

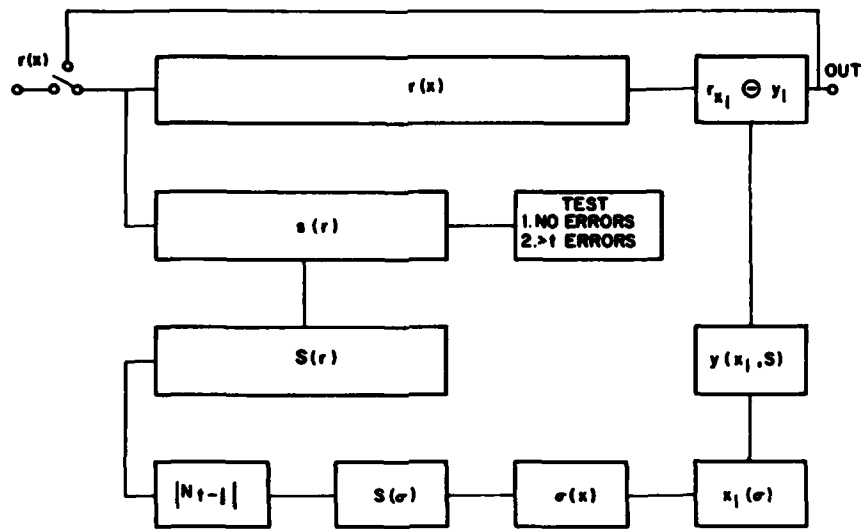


Fig. 6 — Decoding procedure

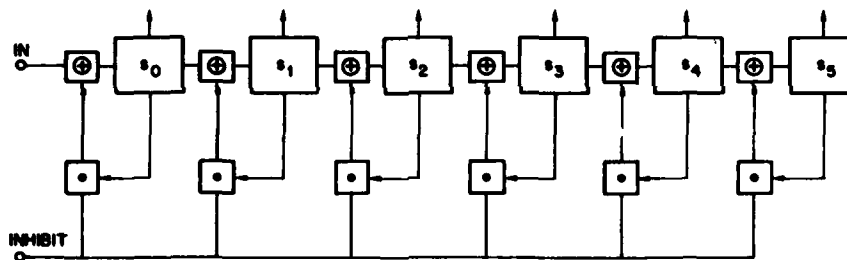


Fig. 7 — Syndrome former

$$(s_0 s_1 s_2 s_3 s_4 s_5) \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 1 & 1 & 1 & 1 & 1 \\ 0 & 0 & 2 & 6 & 14 & 30 \\ 0 & 0 & 0 & 6 & 36 & 2 \\ 0 & 0 & 0 & 0 & 24 & 18 \\ 0 & 0 & 0 & 0 & 0 & 9 \end{pmatrix} = (s_0 s_1 s_2 s_3 s_4 s_5)$$

Fig. 8 — Matrix multiplication for deriving modified syndromes

For $m_0 = 0$, and noting that $0^0 = 1$, the various matrices are related by

$$N_{t-j} = V_{t-j} P_{t-j} V_{t-j}^T,$$

where the distinct root Vandermonde matrix V_{t-j} is defined as follows:

$$V_{t-j} = \begin{bmatrix} 1 & 1 & 1 & \cdots & 1 \\ x_0 & x_1 & x_2 & \cdots & x_{t-j-1} \\ x_0^2 & x_1^2 & x_2^2 & \cdots & x_{t-j-1}^2 \\ . & . & . & . & . \\ x_0^{t-j-1} & x_1^{t-j-1} & x_2^{t-j-1} & \cdots & x_{t-j-1}^{t-j-1} \end{bmatrix}$$

DECODING ALGORITHM

The procedures in the following decoding algorithm are summarized in Fig. 9.

Step 1:

1.0 Define:

$$A_3 = \begin{vmatrix} -S_3 & S_1 & S_2 \\ -S_4 & S_2 & S_3 \\ -S_5 & S_3 & S_4 \end{vmatrix} ; \quad A_2 = \begin{vmatrix} S_0 & -S_3 & S_2 \\ S_1 & -S_4 & S_3 \\ S_2 & -S_5 & S_4 \end{vmatrix} ;$$

$$A_1 = \begin{vmatrix} S_0 & S_1 & -S_3 \\ S_1 & S_2 & -S_4 \\ S_2 & S_3 & -S_5 \end{vmatrix} ; \quad A_0 = |N_3| = \begin{vmatrix} S_0 & S_1 & S_2 \\ S_1 & S_2 & S_3 \\ S_2 & S_3 & S_4 \end{vmatrix} .$$

BECK

For $j = 0$	$-\sigma_3 S_0 + \sigma_2 S_1 - \sigma_1 S_2 = -S_3$ $-\sigma_3 S_1 + \sigma_2 S_2 - \sigma_1 S_3 = -S_4$ $-\sigma_3 S_2 + \sigma_2 S_3 - \sigma_1 S_4 = -S_5$ $-\sigma_3 + \sigma_2 x - \sigma_1 x^2 + x^3 = 0$
For $j = 1$	$\sigma_2 S_0 - \sigma_1 S_1 = -S_2$ $\sigma_2 S_1 - \sigma_1 S_2 = -S_3$ $\sigma_2 - \sigma_1 x + x^2 = 0$
For $j = 2$	$-\sigma_1 S_0 = -S_1$ $-\sigma_1 + x = 0$

Fig. 9 — Algebraic method for locating error positions in code

1.1 If $A_0 = A_1 = A_2 = A_3 = 0$, go to step 2.

1.2 If $A_0 = 0$ and at least one $A_i \neq 0, i = 1, 2, 3$; inconsistent, do not correct.

1.3 If $A_0 \neq 0$, then find

$$-\sigma_3 = \frac{A_3}{A_0}; \sigma_2 = \frac{A_2}{A_0}; -\sigma_1 = \frac{A_1}{A_0}; \text{ and } -\sigma_3 + \sigma_2 x - \sigma_1 x^2 + x^3 = 0.$$

Step 2:

2.0 Define:

$$B_2 = \begin{vmatrix} -S_2 & S_1 \\ -S_3 & S_2 \end{vmatrix}; \quad B_1 = \begin{vmatrix} S_0 & -S_2 \\ S_1 & -S_3 \end{vmatrix}; \quad B_0 = \begin{vmatrix} S_0 & S_1 \\ S_1 & S_2 \end{vmatrix}.$$

2.1 If $B_0 = B_1 = B_2 = 0$, go to step 3.

2.2 If $B_0 = 0$ and at least one $B_i \neq 0, i = 1, 2$; inconsistent, do not correct.

2.3 If $B_0 \neq 0$, then find

$$\sigma_2 = \frac{B_2}{B_0}; -\sigma_1 = \frac{B_1}{B_0}; \text{ and } \sigma_2 - \sigma_1 x + x^2 = 0.$$

Step 3:

3.0 Define:

$$C_1 = |-S_1|; C_0 = |S_0|.$$

3.1 If $C_0 = C_1 = 0$; no errors.

3.2 If $C_0 = 0$ and $C_1 \neq 0$; inconsistent, do not correct.

3.3 If $C_0 \neq 0$, then find

$$-\sigma_1 = \frac{C_1}{C_0}; \text{ and } -\sigma_1 + x = 0.$$

Step 4:

4.0 Substitute b in the appropriate equation, $0 \leq b \leq p-1$, for x , as shown in Fig. 10 (the well-known Chien search), and find the error locations $x_0, x_1, x_2, \dots, x_{t-j-1}$. Go to step 5. If the number of error locations does not equal the degree $t-j$ of the proper equation, more than $t-j$ errors may have occurred; inconsistent, do not correct.

4.1 Alternate Step 4: Solve appropriate equations by algebraic techniques, indicated in Fig. 11, to find error locations.

Step 5:

5.0 Define:

$$|V_3| = \begin{vmatrix} 1 & 1 & 1 \\ x_0 & x_1 & x_2 \\ x_0^2 & x_1^2 & x_2^2 \end{vmatrix}.$$

$$\begin{aligned} &(-\sigma_3, \sigma_2, -\sigma_1, 1) \mathbf{C} \\ &(\sigma_2, -\sigma_1, 1, 0) \mathbf{C} \\ &(-\sigma_1, 1, 0, 0) \mathbf{C} \end{aligned}$$

$$C = \begin{Bmatrix} 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 & 1 \\ 0 & 1 & 2 & 3 & 4 & 5 & 6 & 7 & 8 & 9 & 10 & 11 & 12 & 13 & 14 & 15 & 16 & 17 & 18 & 19 & 20 & 21 & 22 & 23 & 24 & 25 & 26 & 27 & 28 & 29 & 30 & 31 & 32 & 33 & 34 & 35 & 36 \\ 0 & 1 & 4 & 9 & 16 & 25 & 36 & 12 & 27 & 7 & 26 & 10 & 33 & 21 & 11 & 3 & 34 & 30 & 28 & 28 & 30 & 34 & 3 & 11 & 21 & 33 & 10 & 26 & 7 & 27 & 12 & 36 & 25 & 16 & 9 & 4 & 1 \\ 0 & 1 & 8 & 27 & 27 & 14 & 31 & 10 & 31 & 26 & 1 & 36 & 26 & 14 & 6 & 8 & 26 & 29 & 23 & 14 & 8 & 11 & 29 & 31 & 23 & 11 & 1 & 36 & 11 & 6 & 27 & 6 & 23 & 10 & 10 & 29 & 36 \end{Bmatrix}$$

Fig. 10 — Chien search shown in matrix form

For $j = 0$

$$s_0 x^3 + 3s_1 x^2 + 3s_2 x + s_3 = 0$$

$$H = (s_0 s_2 - s_1^2) x^2 + (s_0 s_3 - s_1 s_2) x + (s_1 s_3 - s_2^2)$$

$$(s_0 s_2 - s_1^2) \neq 0$$

$$x = \frac{-(s_0 s_3 - s_1 s_2) \pm [(s_0 s_3 - s_1 s_2)^2 - 4(s_0 s_2 - s_1^2)(s_1 s_3 - s_2^2)]^{1/2}}{2(s_0 s_2 - s_1^2)}$$

Giving x_1, x_2

$$A [(s_0 s_2 - s_1^2) x - (s_0 s_2 - s_1^2) x_1]^3 + D [x - x_2]^3 = 0$$

$$(s_0 s_2 - s_1^2)^3 A + D = s_0$$

$$(-x_1 (s_0 s_2 - s_1^2))^3 A + (-x_2)^3 D = s_3$$

$$\left. \begin{aligned} A &= \frac{\begin{vmatrix} s_0 & b_2 \\ s_3 & c_2 \end{vmatrix}}{\begin{vmatrix} b_1 & b_2 \\ c_1 & c_2 \end{vmatrix}} \\ D &= \frac{\begin{vmatrix} b_1 & s_0 \\ c_1 & s_3 \end{vmatrix}}{\begin{vmatrix} b_1 & b_2 \\ c_1 & c_2 \end{vmatrix}} \end{aligned} \right\} AX^3 = -DY^3$$

For $j = 1$

$$x = 19(\sigma_1 \pm (\sigma_1^2 - 4\sigma_2)^{1/2})$$

For $j = 2$

$$x = \sigma_1$$

Fig. 11 — Algebraic technique for finding error locations

BECK

The error values are:

$$y_0 = \frac{\begin{vmatrix} s_0 & 1 & 1 \\ s_1 & x_1 & x_2 \\ s_2 & x_1^2 & x_2^2 \end{vmatrix}}{|V_3|} ;$$

$$y_1 = \frac{\begin{vmatrix} 1 & s_0 & 1 \\ x_0 & s_1 & x_2 \\ x_0^2 & s_2 & x_2^2 \end{vmatrix}}{|V_3|} ;$$

$$y_2 = \frac{\begin{vmatrix} 1 & 1 & s_0 \\ x_0 & x_1 & s_1 \\ x_0^2 & x_1^2 & s_2 \end{vmatrix}}{|V_3|} .$$

5.1 Define:

$$|V_2| = \begin{vmatrix} 1 & 1 \\ x_0 & x_1 \end{vmatrix} .$$

The error values are:

$$y_0 = \frac{\begin{vmatrix} s_0 & 1 \\ s_1 & x_1 \end{vmatrix}}{|V_2|} ;$$

$$y_1 = \frac{\begin{vmatrix} 1 & s_0 \\ x_0 & s_1 \end{vmatrix}}{|V_2|} .$$

5.2 Define:

$$|V_1| = |1|.$$

The error value is:

$$y_0 = \frac{|S_0|}{|V_1|}.$$

Step 6:

6.0 As a final step, the fully corrected text, together with the corrected parities, is once more entered into the syndrome former for final verification. The new syndrome will now be zero, indicating either the message has been successfully corrected, or miscorrected to another codeword. All other errors at distances greater than d will have been detected in previous inconsistency checks. The complete procedure is outlined in Fig. 6.

SIMPLIFIED SYNTHESIS

In practical code synthesis, the steps shown in Figs. 1, 2, and 3 may be bypassed by first defining S_0^{-1} , which is always in the simple binomial coefficient form of Fig. 5, and then finding its inverse system matrix S_0 , as in Fig. 4, by any conventional means. Further, the system matrix S_0 may be independently found by noting that most of the parity entries are sequential states of a simple nonlinear generator whose coefficients are given by the expansion of $(x - 1)^6$; and the initial segment given directly by the expansion of $(x - 1)^i$, $i = 0, 1, \dots, 6$.

CONCLUSIONS

Several alternatives have been considered for software or hardware implementation of MCJ codes for communication system error protection. It was emphasized that MCJ codes provide improved flexibility over existing procedures in matching an error control code to a given user alphabet size while maintaining optimal performance. Techniques will be presented in the near future, similar to those given here, for synthesis and decoding of Reed-Solomon type codes also over the Galois ground field.

ACKNOWLEDGMENT

Acknowledgment is given to Mr. David C. Andrews who successfully applied his outstanding programming capability to computer modeling of the MCJ codes.

REFERENCES

1. J.L. Massey, D.J. Costello, Jr., and J. Justesen, "Polynomial Weights and Code Constructions," *IEEE Trans. IT-19* (1), 101-110 (1973).
2. R.T. Chien and D.T. Tang, "On Detecting Errors After Correction," *Proc. IEEE* 52, 974-975 (1964).
3. R.C. Bose and D.K. Ray-Chaudhuri, "On a Class of Error Correcting Binary Group Codes," AFOSR Report No. TN 59 1240, September 1959.
4. W.W. Peterson and E.J. Weldon, Jr., *Error-Correcting Codes*, 2nd ed., MIT Press, Cambridge, Mass., 1972.
5. H.M. Beck, "A Sequence Generator for Stirling Numbers of the Second Kind," Report of NRL Progress, January 1976, p. 2.

